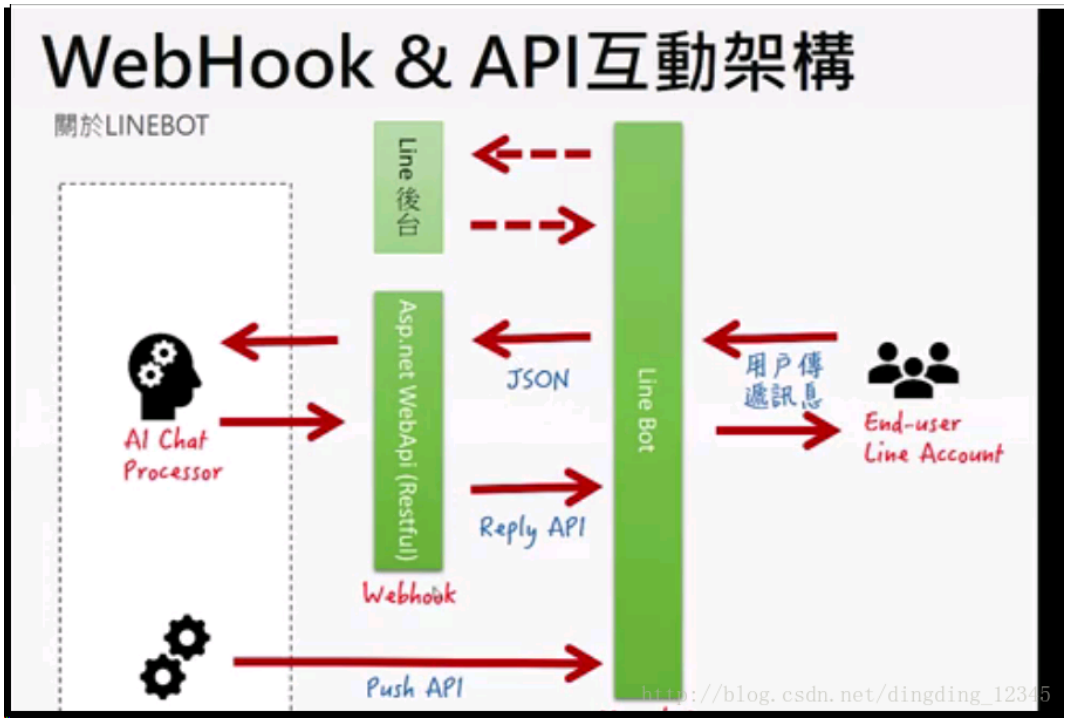
Minestone4

Q1：How is your project architecture related to the theory taught in the lecture?
Our line bot is an example of PaaS, we use Heroku and Google service to provide a platform for users to query the information of epidemic news and situation. We only need to focus on our own business logic, not on the bottom layer. Besides, we use Webhook which is a web callback or http push API, to provide real-time information to line bot. Webhook sends data immediately when the data is generated, which means you can receive the data in real time. This kind of API is different from the typical one. It needs to use real-time and poll fast enough.



The operation process of LINE bot is as follows:
1. The user sends a message to the LINE bot account;
2. LINE bot will post the message to Webhook URL after receiving the message;
3. Webhook URL is what we call web service, responsible for the actual processing of the message.
That is how our project is related to the lecture.

Q2：Can you demonstrate, with some screen cap, how to increase capacity of your chat bot service?

Firstly, there are 4 queries in our chat bot, including the “Epidemic news”, “Epidemic situations”, “Mask sales address”, “nearest hospital”

A. Epidemic news query:
   We created a database in Heroku to store and update the latest news of epidemic, the table including date, news, source. If the users input “epidemic news”, it will

return a flex message of the latest news as the below screenshots show.

After clicking the detail button, the line bot will return the government official website, where the news comes from just like the below screenshot.

B. Epidemic situations query:

Like the query “epidemic”， we also created a table in Heroku to store and update the latest situations of epidemic in Hong Kong, the table including three main districts of Hong Kong including Hong Kong Island, Kowloon, New Territories and its number of infected people. If the users input “epidemic situation”, it will return a flex message of the latest situations of Hong Kong Island, Kowloon, New Territories as the below screenshots show.

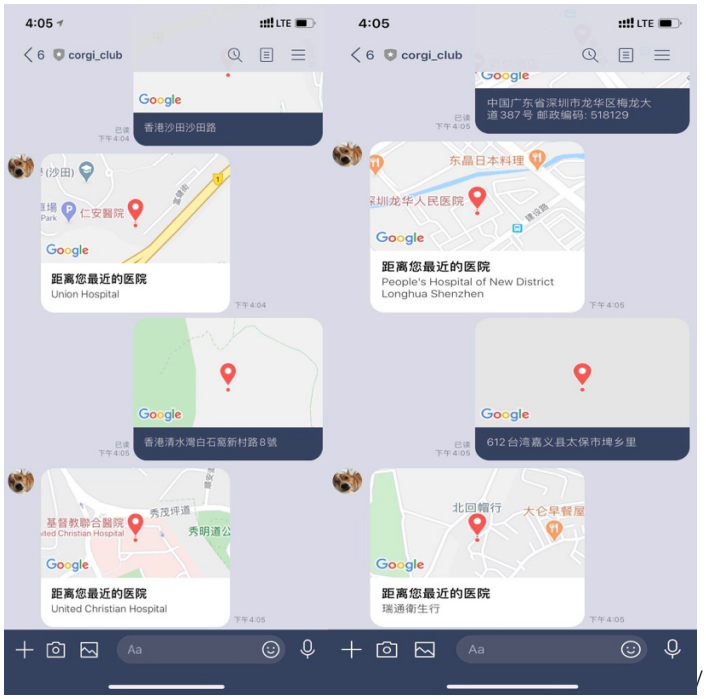
After clicking the detail button, the line bot will return the government official website, where the numbers come from just like the below screenshot.

C. Mask sales address:

Like the query “epidemic”, there is another table named mask store the places which sell masks in Hong Kong. If the users input “mask”, it will return a location message of the places which sell the masks as the below screenshots show.

D. Nearest hospital query

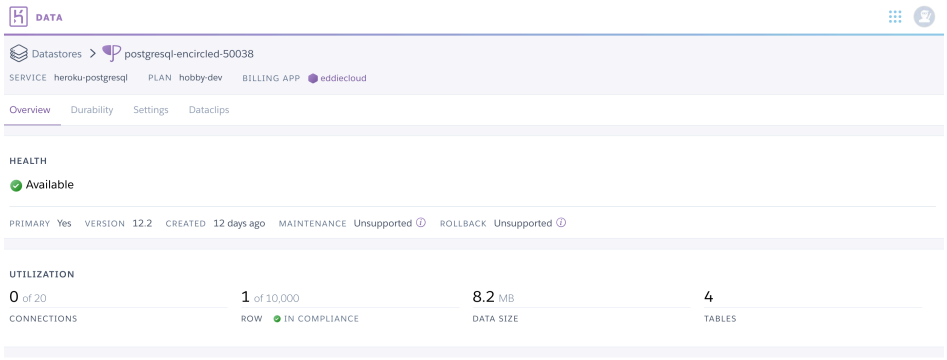
In this part we use google map API to achieve the function that recommends the nearest hospital according to users input location. Here are two screenshots for example.



Besides Redis, we use psycopg2 for solving http request and response to create and modify database in Heroku; google place for recommending the nearest hospital according to users input location. There are three limitations in our line bot project:

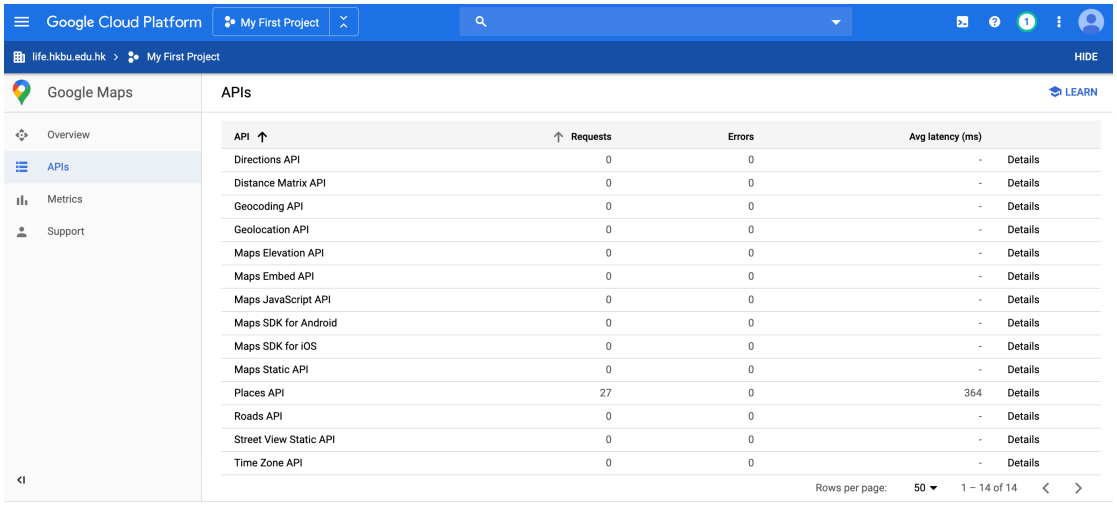
- Limit storage in Heroku database service

  Heroku provide limited storage for programmers, only 10000 rows. If the line bot has more data, it need to increase the volume of database. To overcome this problem, we can try to rent a virtual server and build our own database and backend.



- Limit requests in Google map API

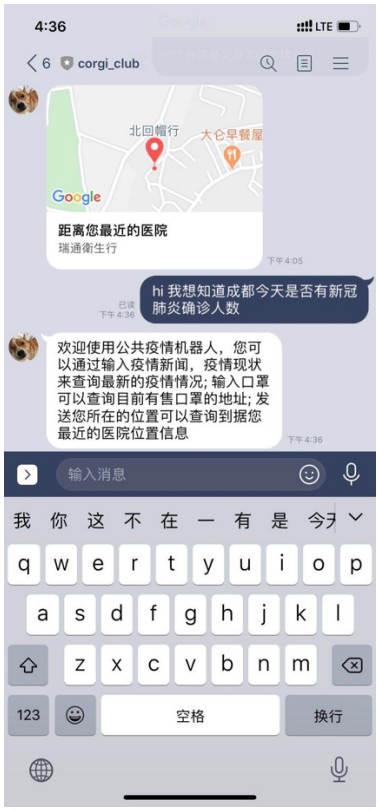
  Google map also provide limited requests from users, If the number of users of line bot has increased a lot, we need to buy a business version API of Google map.



  From the screenshot we can find that, the bot only uses 27 requests and the average latency is 364ms.

- Intelligent query

  In this project we just use a simple string match algorithm, we can only recognize the three strings “epidemic news”, “epidemic situations” and “mask”. If the user input more information, we can’t give him/her a proper response.



To overcome this problem, we can try to use natural language process applications.

Q3：Can you identify if you bot is one of the example of PaaS, IaaS, SaaS? Explain your answer.

PaaS refers to platform-as-a-service, such as DBMS and front-end web server. Users want to extend on-premise environments into the cloud for purposes. Hence, our line bot is an example of PaaS. Just take our project as an example, we use Heroku and Google service to provide a software deployment platform (runtime), abstracts away hardware and operating system details, and can be seamlessly scaled. We only need to focus on our own business logic, not on the bottom layer. Here is a picture of structure of distributed work.